титульник

Оглавление

[Аннотация 3](#_Toc63597834)

[Введение 4](#_Toc63597835)

[Обзор существующих решений 5](#_Toc63597836)

[Общая характеристика 5](#_Toc63597837)

[Иные решения, сравнение 6](#_Toc63597838)

[Теоретическая часть 7](#_Toc63597839)

[Описание работы инженерного решения и рассмотрение алгоритма работы 7](#_Toc63597840)

[Возможность применения на практике 7](#_Toc63597841)

[Способы привлечения средств в проект, описание использованных ресурсов 7](#_Toc63597842)

[Программная реализация 8](#_Toc63597843)

[Описание программы, алгоритм работы программы 8](#_Toc63597844)

[Использованные библиотеки 8](#_Toc63597845)

[Практическая часть 9](#_Toc63597846)

[Список проделанных работ, применение решения на практике 9](#_Toc63597847)

[Заключение 10](#_Toc63597848)

[Список использованной литературы 11](#_Toc63597849)

# Аннотация

Моя работа называется «Предварительная оценка успешности учащихся на курсе».

Как только в мире начал распространяться коронавирус, правительства начали устанавливать меры безопасности, в т.ч. и прямые либо косвенные запреты на передвижение и сборы в компании. Поэтому настал бум всевозможных онлайновых курсов.

Однако, согласно статистическим данным, 90% всех записавшихся пользователей активны только первые 10 дней, после чего либо не активны, либо вообще уходят с курса. Это плохо как для работников курса, так и для самих учащихся. Поэтому данное явление является проблемой, которую призван решить мой проект.

Поэтому, на основе данных за несколько первых дней активности пользователя следует определить, сможет ли он пройти хотя бы половину всего курса, чтобы команда курса могла определить, в зависимости от предпочтений, бросить ли неактивного, или, наоборот, помочь ему. Также можно делать персональные рекомендации плохо успевающим учащимся. Это можно делать при помощи алгоритмов машинного обучения.

Как уже было сказано ранее, наибольшая популярность онлайн-курсов и распространённость вышеописанной проблемы стали выше после начала эпидемии, то есть не более чем год назад. Более того, эта работа нужна для обеих сторон: и для ученика (он сможет получать персональные рекомендации, подсказки или напоминания), и для учителя (он сможет обратить внимание на неактивных учеников, чтобы помочь им, или наоборот, на активных, чтобы заниматься с ними лучше). Значит, проблема актуальна.

Таким образом, моя цель — обучить модель на основании данных по активности пользователя за первую неделю обучения и предсказать с её помощью, пройдёт ли пользователь 50% курса или более, или же нет. Задачи для достижения поставленной цели следующие:

1. Найти подходящий курс с достаточным количеством данных
2. проанализировать данные
3. Извлечь из них необходимые характеристики
4. Обучить на них модель
5. Протестировать, валидировать
6. Получить предсказания

# Введение

Моя работа называется «Предварительная оценка успешности учащихся на курсе».

Как только в мире начал распространяться коронавирус, правительства начали устанавливать меры безопасности, в т.ч. и прямые либо косвенные запреты на передвижение и сборы в компании. Поэтому настал бум всевозможных онлайновых курсов.

Однако, согласно статистическим данным, 90% всех записавшихся пользователей активны только первые 10 дней, после чего не активны либо вообще уходят с курса. Это плохо как для работников курса, так и для самих учащихся. Поэтому данное явление является проблемой, которую призван решить мой проект.

Поэтому, на основе данных за несколько первых дней активности пользователя следует определить, сможет ли он пройти хотя бы половину всего курса, чтобы команда курса могла определить, в зависимости от предпочтений, бросить ли неактивного, или, наоборот, помочь ему. Также можно делать персональные рекомендации плохо успевающим учащимся. Это можно делать при помощи алгоритмов машинного обучения.

Как уже было сказано ранее, наибольшая популярность онлайн-курсов и распространённость вышеописанной проблемы стали выше после начала эпидемии, то есть не более чем год назад. Более того, эта работа нужна для обеих сторон: и для ученика (он сможет получать персональные рекомендации, подсказки или напоминания), и для учителя (он сможет обратить внимание на неактивных учеников, чтобы помочь им, или наоборот, на активных, чтобы заниматься с ними лучше). Это подтверждает актуальность проблемы. Так что мой проект призван облегчить работу учителям курсов, а заказчики — работники платформы курса.

Таким образом, моя цель — обучить модель на основании данных по активности пользователя за первую неделю обучения и предсказать с её помощью, пройдёт ли пользователь 50% курса или более, или же нет. Задачи для достижения поставленной цели следующие:

1. Найти подходящий курс с достаточным количеством данных
2. Проанализировать данные
3. Извлечь из них необходимые характеристики
4. Обучить на них модель
5. Протестировать, валидировать
6. Получить предсказания

Данное решение представляет собой модель типа «случайный лес», обученную на следующих параметрах: Сколько баллов пользователь набрал за первую неделю, сколько задач решил правильно и неправильно, сколько комментариев оставил за неделю и (целевой признак, в обучение не включался) смог ли он за всё время обучения на курсе набрать хотя бы 50% от общего количества баллов. Точность предсказаний по метрике оперативной характеристике приёмника должен составлять не менее 0,88.

Технической составляющей данного решения как бы и нет, так как оно является программным модулем и, теоретически, может быть исполнено на любом компьютере с достаточными техническими требованиями (очень быстро работала модель на компьютере с процессором из 6 ядер частотой 3,17 ГГц, ОЗУ 16 ГБ) и всеми средствами, указанными далее в разделе «описание платформ и библиотек».

# Обзор существующих решений

## Общая характеристика

Данное решение представляет собой модель типа «случайный лес», обученную на следующих параметрах: сколько баллов пользователь набрал за первую неделю, сколько задач решил правильно и неправильно, сколько комментариев оставил за неделю и (целевой признак, в обучение не включался) смог ли он за всё время обучения на курсе набрать хотя бы 50% от общего количества баллов.

Чтобы объяснить, как работает «случайный лес», надо знать некоторые базовые принципы.

В данном случае мы сталкиваемся с задачей классификации — распределить объекты по конечному множеству классов по их признакам (эта задача относится к задачам обучения с учителем — нам даются независимые переменные-«признаки» и «желаемые ответы», т.е. имеется некий «учитель», «наставляющий» модель). В данном случае наипростейшим алгоритмом является классификатор на основе дерева решений — дерева, в узловых вершинах которого находятся решения (по типу «переменная больше или равна значению»), а в листьях — окончательные решения по распределениям объектов в классы, это можно сравнить с тем, как человек по признакам животного определяет, кошка это или собака. Алгоритм обучения дерева умеет «задавать правильные вопросы» (т.е. подбирать такие разбиения по значениям, чтобы они давали наибольший выигрыш в полученной информации), так что дерево обучается практически безошибочно. Однако если заглянуть в структуру дерева, просто бесконтрольно обученного на неком датасете, можно увидеть, что оно разрослось и «додумалось» до каких-нибудь невообразимых комбинаций, а значит, переобучилось — нашло какие-то закономерности в тренировочных данных, которых может не быть в тестовых. Но у дерева есть набор начальных параметров — максимальная глубина и т.д. То есть мы, регулируя параметры, можем изменять степень переобученности.

Но дерево всегда будет немного переобученным (такова природа алгоритма — он по своей сути ищет закономерности и либо недобирает нужных, либо нахватывается неправильных), поэтому была придумана следующая модель — случайный лес. Он является совокупностью деревьев решений с одинаковыми начальными параметрами, каждое из которых обучается на случайном фрагменте тренировочного датасета. Затем, при применении, эти деревья каждое делают предсказания на весь тестовый датасет, их предсказания усредняются и выводятся как окончательный вердикт. В целом получается достаточно точно — это можно сравнить с тем, когда у людей на улице спрашивали — сколько весит средний бык? Мало кто ответил правильно, но их усреднённые ответы были очень близки к истине.

## Иные решения, сравнение

К сожалению, иные решения подобных вопросов мне неизвестны, но я предполагаю, что, скорее всего, крупнейшие платформы онлайн-курсов (такие, как Stepik, Coursera и т.д.) уже создали аналогичные модели и успешно их применяют. Сравнение их с моим решением не представляется возможным, так как они, скорее всего, скрыты внутри сервисов и доступ конечного пользователя к ним невозможен, а обратная разработка данного вопроса, скорее всего, противозаконна (потому что найденные решения, весьма возможно, принадлежат сотрудникам курсов, зарегистрировавших свои решения как интеллектуальную собственность) или невозможна, как сказано выше.

Преимущества и недостатки также не могут быть выявлены по тем же самым причинам.

# Теоретическая часть

## Описание работы инженерного решения и рассмотрение алгоритма работы

Предложенное мной решение не является техническим устройством, поэтому детальное описание алгоритма работы и его самого находится в разделах «общая характеристика» и «программная реализация».

## Возможность применения на практике

Полученные при помощи обученной модели данные можно использовать во благо учеников (посылать им рекомендации, подсказки, если он надолго застрял, напоминания и т.д.) и учителей (чтобы указать им на тех, кто работает интенсивно и на тех, кто плохо справляется). Также по мере поступления новых данных модель следует дообучать, чтобы она не «теряла хватку», оставалась в курсе новых намечающихся закономерностей и т.д.

Саму модель можно встроить во внутреннюю, скрытую от конечного пользователя, часть сайта, но для этого нужно либо получить разрешение от владельцев платформы, либо создавать свою курсовую платформу, поэтому пока данное решение существует отдельно.

## Способы привлечения средств в проект, описание использованных ресурсов

В качестве среды разработки был использован Visual Studio Code и дополнение Jupyter Notebook, отчёты с одного из курсов на Stepik, библиотеки pandas, numpy, sklearn и seaborn.

Проект основан на свободно распространяемых ресурсах, пока не зарегистрирован как чья-либо интеллектуальная собственность, поэтому сам абсолютно свободен, бесплатен и доброволен. Репозиторий на Github: <https://github.com/MichYar/GreatChallenges-2020>.

Список проведённых работ приведён далее, в разделе «Список проведённых работ и применение решения на практике». Более подробный список проведённых работ приведён в виде списка задач проекта и на приведённом далее рисунке 1.

# Программная реализация

## Описание программы, алгоритм работы программы

Описание программы сводится к последовательности инструкций, указанных в качестве задач проекта или к заголовкам больших контейнеров (блоков, объединяющих комбинации действий в некий единый смысловой блок) на приведённом ниже рисунке 1.

Здесь я привожу процесс подготовки данных и то, как пользователь видит обучение машины (оно, как и многие другие вещи, скрыто и является чёрным ящиком) в виде блох-схемы средней степени подробности (то есть не сводящейся к общим инструкциям, но и не описывающую полную систему библиотечных функций) (рис.1). Большая часть процессов помечена как «предопределённый процесс», так как являются либо библиотечной функцией-«чёрным ящиком», либо комбинацией таких функций. Файл блок-схемы также доступен в векторном формате SVG.

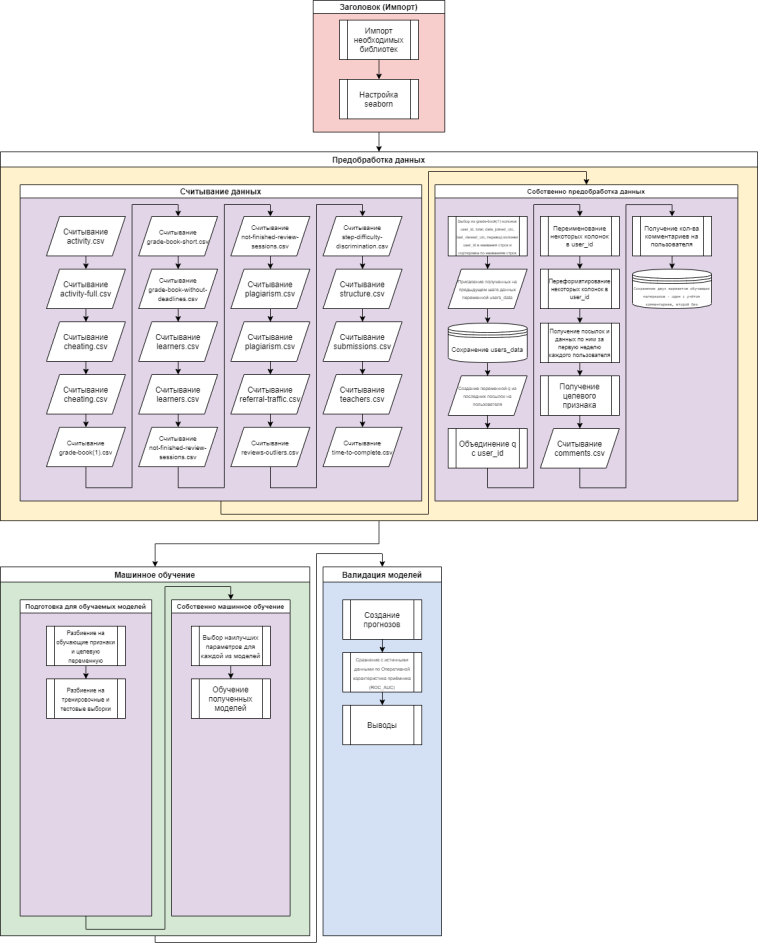


Рисунок 1. Блок-схема алгоритма.

Полная версия программы доступна в репозитории на Github, с комментариями на русском и английском языке.

## Использованные библиотеки и платформы

* Visual Studio Code. Настраиваемый редактор кода для большинства типов файлов и языков. Многофункционален благодаря обширной базе расширений.
* Python 3.9.1. Языковая платформа, на которой писался проект, и расширение для VS Code для работы с ним.
* Jupyter Notebook. «Блокнот» для работы с Python, визуализациями и форматированным текстом (реализовано при помощи Markdown) в одном файле и расширение для VS Code для работы с Notebook прямо в редакторе.
* Pandas 1.2.1. Библиотека для работы с данными.
* NumPy 1.19. Библиотека для работы с высокоуровневыми функциями для многомерных массивов данных, обязательное требование pandas и SKLearn.
* SciPy 1.6.0 Библиотека для инженерных расчётов. Обязательное требование для SKLearn.
* Matplotlib 3.3.3. Библиотека для визуализации графиков, обязательное условие для seaborn и SKLearn.
* Seaborn 0.11.1. Надстройка для matblotlib, позволяет строить улучшенные графики.
* SKLearn 0.24.1 Библиотека, реализующая алгоритмы машинного обучения, метрики качества обучаемых моделей и т.д.

# Практическая часть

## Список проделанных работ и применение решения на практике

1. Обучение способам машинного обучения
2. Работа на целевой проект

В первой работе я проходил курс обучения с созданием похожей модели с задачами, подобными задачам текущего проекта. Создавалась она примерно также, как и текущее изделие, за исключением различий в способах обработки данных. Эта модель создавалась исключительно в целях обучения работника основным принципам машинного обучения и работы с данными, и эта модель единственная, которая использовалась практически (её предсказания были посланы в тестирующую систему и оценены).

Вторая работа — мой текущий проект. Этот проект является модификацией старого относительно нового формата данных и способа обработки этих данных, но идеологически он почти не отличается от предыдущего. К сожалению, моя модель пока не применялась на практике серьёзно (только оценивалась на существующих данных). Он должен пройти дополнительную проверку, что пока невозможно из-за малого количества новых данных, и одобрение заказчика на его внедрение.

# Заключение, выводы

Данное решение является моделью вида «классификатор на основе случайного леса» для обеспечения знания о возможных успевающих и неуспевающих учениках онлайн-курса. Данная модель создана для применения на практике, но пока не может из-за недостаточной степени проверенности и валидированности модели. Скорее всего, созданы модели подобной направленности и, возможно, даже более хорошего качества, но, как уже было сказано ранее, изучить эти модели не представляется из-за недоступности их для конечных пользователей и незаконности процесса обратной разработки данных систем. В процессе работы над этим решением я научился основным принципам так называемого Data Science (англ. Наука о Данных, или данные как наука), т.е. обработке данных, извлечению из них нужной информации, обучению на ней машинных моделей для получения предсказаний и т.д. Так что этот проект был полезен не только конечному заказчику, но и работникам, вовлечённым в проект.

Достигнута довольно высокая точность (оперативная характеристика приёмника наилучшей из моделей, протестированной на всех данных — 0,988):

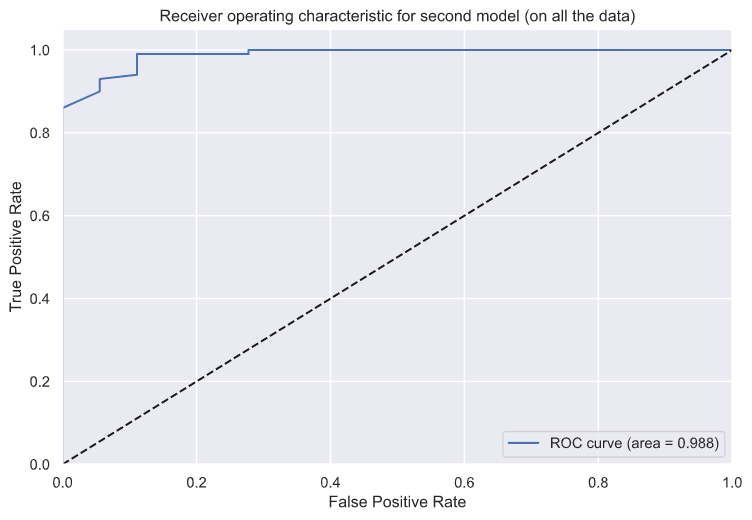


Рисунок 2. График оперативной характеристики приёмника.

Этот результат вполне соответствует той точности, к которой я стремился.

Методика испытаний заключалась в тестировании на существующих данных, поэтому испытания модели не могут считаться завершёнными по причине отсутствия достаточного количества новых данных.

# Список использованной литературы

1. Введение в Data Science и машинное обучение [Электронный ресурс], URL: <https://stepik.org/course/4852/syllabus> (Дата последнего обращения: 31.01.2021)
2. Pandas Documentation [Электронный ресурс], URL: <https://pandas.pydata.org/docs/> (Дата последнего обращения: 06.02.2021)
3. SKLearn API Reference [Электронный ресурс], URL: <https://scikit-learn.org/stable/modules/classes.html>  
   (Дата последнего обращения: 04.02.2021)
4. Python 3.9.1 Docs [Электронный ресурс], URL: <https://docs.python.org/3/> (Дата последнего обращения: 15.01.2021)